

## Contents

<b>1 Routine/Function Prologues</b>	<b>2</b>
1.0.1 interp_agrmet_sw.F90 (Source File: interp_agrmet_sw.F90) . . . . .	2

## 1 Routine/Function Prologues

### 1.0.1 interp\_agrmet\_sw.F90 (Source File: interp\_agrmet\_sw.F90)

Opens, reads, and interpolates AGRMET shortwave radiation forcing

#### REVISION HISTORY:

```

26 Jun 2001: Urszula Jambor; Initial code, based on Jesse Meng's
              RTNEPH2LATLON.F code.
08 Feb 2002: Urszula Jambor; Modified declarations of arrays
              dependant on domain & resolution to allocatable.
              Pass in values for latmax.
11 Dec 2002: Urszula Jambor; Added 1/2 & 1 degree resolution GDS arrays
04 May 2004: James Geiger; Added opendap support through lis_openfileMod
              and lis_indices_module

```

#### INTERFACE:

```
subroutine interp_agrmet_sw( nameSH, outdata, ferror )
```

#### USES:

```

use lisdrv_module,only : lis, gindex
use agrmetdomain_module, only : rlat,rlon,w11,w12,w21,w22,n11,n12,n21,n22,mi,mo
use lis_openfileMod
use lis_indices_module
implicit none

```

#### ARGUMENTS:

```

character*80 :: nameSH
real :: outdata(lis%d%ngrid)
integer :: ferror

```

#### CONTENTS:

```

allocate(pdata(mi))
allocate(lo1(lis_nc_working*lis_nr_working))
allocate(ldata1(lis_nc_working*lis_nr_working))
print*, 'Reading AGRMET file : ',nameSH
call lis_open_file(11, file=nameSH, form='unformatted',script='getagrmet_sw.pl')
read(11, iostat=readerrN) pdata1
close(11)

if ((openerrN+openerrS+readerrN+readerrS) > 0) then
  ferror = 0
  if ((openerrS+readerrS) > 0) then
    print*, 'AGRMET file problem:', nameSH
  end if
  do i=1,lis%d%ngrid

```

```
        outdata(i) = lis%d%udef
      end do
    else
      ferror = 1
      ibi = 1
      count = 0
      li1 = .false.
      do j=1,nagrr
        do i=1,nagrc
          pdata(count+i) = pdata1(i,j)
        enddo
        count = count+nagrc
      enddo
      do i=1,mi
        if(pdata(i).eq.-9999) then
          li1(i) = .false.
        else
          li1(i) = .true.
        endif
      enddo
      kgdso = 0
      kgdso = lis%d%kgds
      call polates0(kgdso,ibi,li1,pdata,ibo,lo1,lidata1,mi,mo,&
                    rlat,rlon,w11,w12,w21,w22,n11,n12,n21,n22,iret)

      if(iret .NE. 0) then
        print*, "IPOLATES ERROR!! PROGRAM STOP!!"
        call exit(iret)
      end if
      count = 0
      do j=1,lis_nr_working
        do i=1,lis_nc_working
          if(gindex(i,j).ne. -1) then
            outdata(gindex(i,j)) = ldata1(i+count)
          endif
        enddo
        count = count+lis_nc_working
      enddo
    endif
    deallocate(pdata)
    deallocate(lo1)
    deallocate(lidata1)
```